**Three New Ways Used to Improve the Data**

1. Feature Engineering: Created a new feature called 'total_guests' which is a sum of 'children' and 'babies', potentially capturing family bookings better.

2. Handling Missing Values: Filled missing values with the median, which is robust to outliers and can improve model accuracy.

3. Normalization: Applied MinMax scaling to numerical features to ensure that all features contribute equally to the result.

**Error Analysis and Data Improvements**

After the initial model training, error analysis might reveal patterns in the mistakes the model is making. For instance, if the model frequently misclassifies a certain class, I might consider oversampling that class or creating new features that help distinguish it better.

**Performance Comparison**

I calculate metrics such as AUC, accuracy, precision, recall, and F1-score on the validation dataset for both the Week 9 model and the updated model. Compare these metrics to see which model performs better on the validation dataset. The model with the higher AUC on the validation set is considered to have performed better. If the updated model has a higher AUC, it's likely because the data improvements allowed the model to capture the underlying patterns in the data more effectively, reducing overfitting and increasing its generalization capabilities.

**Selecting the Final Model for Deployment**

Choose the model with the better performance metrics on the validation dataset as my final model for deployment. This choice is based on the model's ability to generalize to unseen data while minimizing the chance of future performance degradation. Evaluate the final model on the test dataset to get performance metrics. These metrics are critical because they give me the final verification of how my model is expected to perform in the real world.

Ideally, the test error should be comparable to the validation error, which would indicate that the model is generalizing well. A much higher test error could suggest that the model is not as robust as the validation error indicated.

**Insights Based on Errors**: If the training error is significantly lower than the validation and test errors, the model might be overfitting. If all errors are high, the model might be underfitting. Consistently low errors across all datasets suggest a well-fitting model. A close examination of where the model makes errors can guide further data preprocessing or feature engineering. The final selection between the models should also consider the complexity of the model, the interpretability, and the operational costs of using the model, such as inference time and resource requirements.